# LOADSTAR LETTER

## Number 32. Companion the LOADSTAR disk #142

## Just when You Thought You Were "Compleatly Safe," LOADSTAR Launches Two New Software Titles


LOADSTAR proudly introduces two new titles in our *Compleat* series: *The Compleat Crossword* and a new game disk, *The Compleat Jon. The Compleat Crossword* is a major collection of 220 crossword puzzles found in *Puzzle Page* on LOADSTAR.

From a handy scrolling menu you can select any of the puzzles and then solve them on screen with Barbara Schulak's *Cruciverbalist* program.When done with the puzzle you have the option of "marking" the puzzle as one that's solved.

*Compleat Crossword* is a testament of digital cooperation, compiled by Bill Calvert, Presenter by Fender Tucker, and Puzzle Program by Barbara Schulak. The myriad of puzzles were crafted by Barbara ~~Schulak, Fender Tucker, Craig Buchman, Linda~~ Vano, Peter Rokitski and Knees Calhoon Music by Dave Marquis. You can even select which of the nine songs you'd like to hear while solving the puzzles.

*The Compleat Jon,* The Games of Jon Mattson, is a collection of 11 superb games previously published on LOADSTAR. During LOADSTAR's "middle years" (1987-1990) one programmer stood out from the crowd because of the sophistication of his programs -- Jon Mattson. Practically every month he would send LOADSTAR his latest creation, which usually used the most modern techniques of game programming.


The whole gamut of gaming is covered here: artificial intelligence, role-playing, mazes, fantasy, science fiction, education and even non-violence (which was a radical concept in its time).

The thing I like best about the games is that each takes place in a consistent, realized world of its own. Read the docs, which are all by Jon himself, to get the background story behind each game.

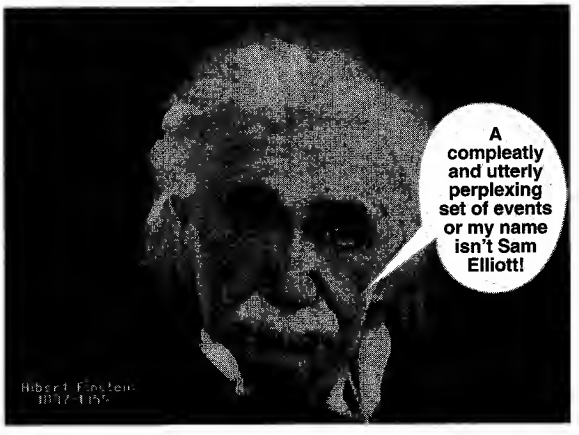
These are not silly little games made with character graphics; they are games with backgrounds and stories, and Jon tells these stories very well, indeed. By the time you finish the docs, you are *in* the created world, and once you run the program you *are* the hero, with lives depending on you and your cleverness and dexterity.

## Things That Make You Go "Hmmm..."

by Jeff Jones.While developing *The Compleat Crossword*, Fender stumbled upon a bug in my [Jeff} code that causes a crash whenever following 6502 code creates an integer variable, F%, with a value of 40.

```
;lsb in 253
;msb in 254
lda "f"
sta 251
lda "%"
sta 252
lda $7a
pha
lda $7b
pha
lda <251
sta $7a
lda >251
sta $7b
jsr $b08b
ldy #0
lda 254 ;high byte
sta ($47),y
iny
lda 253 ;low byte
sta ($47),y
pla
sta $7b
pla
sta $7a
rts
```

This code was handed down to me from C-64 programming legend, Rick Nash. I'll admit that I've mutated the code a bit. I use this code all the time to pass data back to BASIC from my machine language routines. I used this code in *Menu Toolbox*, which Fender used in *Compleat Crossword*. When Fender selected the 40th item on the crossword menu, the program crashed with a syntax error. I studied the code and found nothing that should be agitated by a value of 40. I plugged in a 40 and assembled the code by itself. *Crash!* What's more weird, Toolbox 105 uses similar code with different memory locations to protect its variables. It *doesn't* crash on 40. I can't figure out why. I was able to fix the code by making the variable zero, and then manually plugging in the integer value, using the $5f pointer to the variable data.




I found the pointer by locating a variable in memory, and then using Super Snapshot V5.5.2's machine language monitor to search for any zero page locations pointing to the variable. Though this bug is squashed, I can't help but wonder what could have caused such a thing to happen. Hmmm. Anyone have any ideas?

 LOADSTAR LETTER contacts: INTERNET: LOADSTAR@GENIE.GEIS.COM GEnie: LOADSTAR US MAIL: ATTN Jeff Jones, J & F Publishing P.O. Box 30008, Shreveport, LA 71130-0008 318/221-8718. Fax 318/221-8870. BBS 1-318-425-4382. Leave feedback or email "Jeff Jones" 2400/8n1. N.I.S.S.A. Network

# LOADSTAR® MONTHLY

LOADSTAR is a monthly "magazine on disk" for the Commodore 64/128. Subscribers receive two 1541 disks (or one 1581 disk) in their mailbox every month packed with news, articles and programs. These non-PD, high-quality programs are written by the best home-based programmers in the field and edited by the crack LOADSTAR team of Fender Tucker and Jeff Jones. Subscription prices are at an all-time low of $69.95 for a 12-month subscription, or $19.95 for a three-month subscription. You may also elect to subscribe "by the month," where we charge your credit card $7.95 for each issue after it's shipped. We also offer the long line of standalone products below.

✔ **NEW Games Disk! The Compleat Jon:** 11 Games! The whole gamut of gaming is covered here: artificial intelligence, role-playing, mazes, fantasy, science fiction, education and even non-violence (which was a radical concept in its time). These eleven games are among the best ever published on LOADSTAR. Listed on the menu in chronological order, so you can see how Jon's style changed as the years rolled by. **1581 disk 0021D3 $20. 1541 disk #0038D5 $20**


✔ **NEW Puzzle disk! The Compleat Crossword:** Every crossword puzzle published in Puzzle Page in one huge collection! 220 puzzles! It uses Barbara Schulak's CRUCIVERBALIST program to present the puzzles and allows you to "mark" a puzzle when it's solved so that you know which you've solved and which you haven't yet. Each 1541 disk contains 110 puzzles. **1581 Disk #0020D3 $20. Disk 1 (1541) #0036D5 $10.Disk 2 (1541) #0037D5 $10**

✔ **NEW C-128 Productivity! The Compleat Lee O:** Six of Lee O. Clinton's best serious programs for the C-128 80- column mode. Finance, auto expense, kitchen helper, genealogy, resume writing, mutual funds! **One 1541 disk #0032D5. One 1581 disk #0017D3 $10.00**

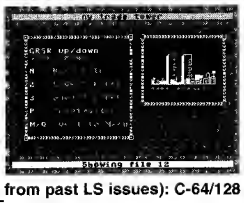
✔ **NEW Word Search! Super Star Search 1:** 200 original word search puzzles by Steven Thomas and Art Dudley, presented by John Serafino's modern point and click program. **One 1541 disk #0011D5 One 1581 disk #0008D3 $20.00**

✔ **NEW Story disk! The Compleat Prosequest '95:** NEW!!! A 1541 disk with all of the entries in the 1995 short story writing contest on it, including the three grand winners. **One 1541 disk #0035D5 One 1581 disk #0019D3 $5.00**

**Soft Wear! LOADSTAR T-Shirts:** Limited edition Fruit Of The Loom T-shirts. LOADSTAR's nemesis, Knees Calhoon, stands up to regular washing and drying. Where else can you find Commodore apparel? 50% Cotton//Polyester. **$15.00 each with free shipping! Small #960025, Medium #960125, Lsrge #960225, X-Lsrge #960325, XX-L #960425**

**Tools! Tutorials! The Compleat Programmer:** Best seller! Over Two megabytes of knowledge crammed and stuffed onto eight 5.25-inch disks or two 1581 disks! Plus tools, extensions, languages, assemblers, tutorials and utilities! **5.25 set #0005D5. 3.5-inch disk #0005D3. For $5 more, get C= Hacking MAG #0006D3 (on 3.5-inch disks only snd NOT availsble separately)** to complete your programming set. **$20.00**

**PS Graphics! Compleat PS Vol. 1** (The Print Shop by Broderbund or Printmaster required). Over 1300 artistic and never before published PRINT SHOP images. Scan through the many PRINT SHOP images sequentially, by name, or by group number. Press a key and save the graphic you want in 2-block, 3-block and even PRINTMASTER graphic files! All that plus a printed guide! Each volume is $20.00. **Vol. 1: C-64/128 3.5-inch disk item #0001d3. 5.25-inch disks item #0009d5. Vol. 2 (grsphics from past LS issues): C-64/128 3.5-inch disk item #0002d3. 5.25-inch disks item #0010d5.**



## GEOS Clipart

LOADSTAR presents the biggest Geos collection of clip art and fonts ever offered at one time. All of the Geos art that's ever appeared on LOADSTAR, as well as some great files from Geos fanatic Dick Estel, are available on twenty 5.25 inch disks or eight 3.5 inch disks. Most of this has never been seen before! Use these graphics in your GeoPaint, GeoWrite and GeoPublish documents or convert to FGM with FGM utilities. Spiff up your GeoFAX documents with the appropriate graphic -- every time! Prices are $20 for any two 3.5 inch disks, or any five 5.25 inch disks. You can purchase the whole collection for $75 for either version. Call LOADSTAR toll-free at 1-800-594-3370 or 1-318-221-8718 to order by credit card. Or send check or money order and specify (by LG number) which disks you want.

### 5.25-INCH DISKS

**Disk 01 - RAILS: Railroad art** from Europe and the USA **#0012D5**

**Disk 02 - VEHICLES/TAROT:** Artwork of old and new autos; excellent geoPaint drawings of the Tarot card set **#0013D5**

**Disk 03 - CLIP ART:** Includes converted MacPaint files that have never before been available in Commodore format **#0014D5**

**Disk 04 - OTTOWA/PRIME CLIPS:** Artwork of the main landmarks of Ottowa; plus high quality public domain clip art **#0015D5**

**Disk 05 - FONTS:** More than 30 fonts from past issues of LOADSTAR, plus articles (in geoWrite format) on creating fonts. Also two ready-made headers for use with your own documents, one a picture of a mail truck; the other reading FROM THE DESK OF **#0016D5**

**Disk 06, Disk 07,** and **Disk 08** - geoPaint and Photo Album files with the great clip artwork featured on past LOADSTARS - Includes GeoCurmudgeon, Anamalia I and II, Australian Animals, Valentine art and many more **#0017D5, #0018D5, #0019D5**

**Disk 09 - GOODYKOONTZ FILES** - Jasper Goodykoontz, born in Indiana in 1855, produced Goodykoontz's Perpetual Calendar and General Reference Manual (A Book for the Millions). This disk includes scans from the book of a wide array of subjects -- Gestures and Attitudes, Poultry, Craniology, and more. **#0020D5**

**Disk 10 - OLD WEST:** Scanned Artwork from Dick Estel's FRD Software - mostly woodcut style art of the old west, gold rush days and pioneer scenes. **#0021D5**

**Disk 11: J. Neely Art/Animals:** Jennifer Neely works with a wide variety of subject matter and materials. Disk contains some of her favorites, scanned into geoPaint format. Side 2 is a collection of scanned artwork of animals from FRD Software **#0022D5**

**Free Shipping!** For Every item on this page!

**Disk 12 - HOLIDAY:** Artwork for New Years, Valentine's, St. Patrick's Day, Halloween, Thanksgiving and Christmas **#0023D5**

**Disk 13 - PEOPLE/FACES:** Scenes of people and faces from FRD Software **#0024D5**

**Disk 14 - FRD CLASSICS:** Dick's choice of the best of the FRD collection **#0025D5**

**Disk 15 - DINOS/CLASSICS:** Dinosaurs and other prehistoric beasts, as well as more first choice artwork from FRD. **#0026D5**

**Disk 16 - SPORTS/MISC:** Dozens of sports-related clips **#0027D5**

**Disk 17 - OFFICE AND SCHOOL:** Clips to be used at work and around the house **#0028D5**

**Disk 18 -MUSIC & MORE SCHOOL CLIPS #0029D5**

**Disk 19 - SEASONAL AND HOLIDAYS:** A clip for any occasion **#0030D5**

**Disk 20 - SEASONAL AND HOLIDAYS:** A clip for any occasion **#0031D5**

### 3.5 INCH DISKS

The 3.5" disks are roughly equivalent to two and a half 5.25" disks.

**Disk 1: Equals disks 1, 2, 4B #0009D3**
**Disk 2: Equals disks 3, 6, 7A #0010D3**
**Disk 3: Equals disks 5, 8, 7B #0011D3**
**Disk 4: Equals 9, 10, 11A #0012D3**
**Disk 5: Equals 12, 13, 11B #0013D3**
**Disk 6: Equivalent of Disks 14, 15** and some bonus files not on 5.25" disks **#0014D3**
**Disk 7: Sports, Office and school, Music #015D3**
**Disk 8: Music, Holiday and Seasonal #016D3**

For your convenience, GeoViewer is included on each volume. GEOS 2.0 is suggested.

**Diskfulla Card Games! The Compleat Maurice:** A compilation of 26 solitaire card games written by Maurice Jones, the acknowledged master of card game simulations for the C-64/128. There's even a brand new, never before published game called Boomerang. **Two 5.25 inch disks #0007D5 or one 3.5 inch disk #0007D3. $20.00 postsge paid!**

**Oodles Of Stunning Art! Tutorials! The Compleat Walt:** During LOADSTAR's first ten years we have published 24 of Walt Harned's slideshows and multimedia events. Now we've gathered them into one huge collection: seven 5.25 inch disks or three 3.5 inch disks! There are over 250 pictures, including some that have never been published. The greatest one-man collection of art on any computer platform. **5.25-inch disks order #070425 3.5-inch disks order #070423. $20.00 postage paid!**

# Loadstar Order form ☎ 1-800-594-3370

| QTY | Description | Item # | Price ea. | Total |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |

Use extra sheet of paper for large orders

Name _____

Address _____

City _____ State _____ Postal Code _____

**Total Enclosed:** _____

❑ Check/money order made payable to "Loadstar" in US Funds

❑ MasterCard ❑ Visa ❑ American Express ❑ Discover

Card # _____

Exp Date ____/____ Authorized signature: _____

Subscribers: I prefer ❑ 5.25-inch diskettes ❑ 3.5-inch diskettes

**LOADSTAR**
**P.O. Box 30008, Shreveport, LA 71130-0008**
**Questions: 1-318-221-8718**

# Xnet, The Newest Commodore BBS Network

If you are currently running an Image BBS and want to join a network then look no further! JOIN XNET!! Just leave feedback (not Email) on the LOADSTAR BBS at 1-318-425-4382, saying that you wish to join! We would love to have you in our network!! No harsh rules! Just a bunch of Sysops having FUN!

# Jeff's Eight Inches

On the LOADSTAR LETTER #25, I slammed the STAR SJ-144 color printer. I got one letter in its defense last fall. For the way the writer used the printer (a cool color graphic covering 20% of the paper in the middle of text), it seemed to have decent output. It still doesn't stack up to modern color ink jet printers. The Star SJ-144 isn't history because Jeff doesn't like it. It's history because it doesn't stack up. And how dare they call it "laser quality." It's hard enough to keep a straight face when you call a serious ink jet printer "laser quality."

Frank Vanaman wrote me, informing me that the Casio digital camera's 320x200 resolution wasn't sufficient for stunning photos, and that it was unfair to compare it with the scan of Fender and Judy in the LOADSTAR T-shirt ad. He also informed me that the 16 million color photos can make a photo look real at first glance, but at 320x200 pixels, you can still see the lack of detail.

I agree. Also one thing some people don't understand about "16 million color pictures" is that a 320x200 image only has 64,000 pixels, and if each were a different color, it would only be a 64,000 dot picture. THere are 16 million colors *available* in a 24-bit picture. I just left Kroger where I perused a Modern photography. They asked if digital photos would overtake 35mm. They did side-by-side comparisons of digital cameras, including the Casio QV-10, and *all* the digital cameras paled to old fashioned 35mm cameras. To ad insult to injury, they took a picture with a lowly disposable camera, then scanned it, and came up with a better picture!

But there was a line of professional digital cameras that ranged from a few thousand to $27,000, which took beautiful sharp photos at more than 1000x1000 pixels. I was impressed. Maybe in ten years, I can get a camera like that for a few hundred. Until then, digital artists with non-corporate budgets will continue to rely on "real" film cameras. I'm looking at a good focusable Polaroid camera for use with my scanner.

While editing a program this month, I came upon an interesting way to mix Commodore uppercase graphics with Upper/lowercase text on a Commodore printer. Open two channels to the printer. Open4,4,7:open5,4,0. Print#4 for text and print#5 for graphics. Why had I never thought of it?

I got a piece of Email, stating that LOADSTAR pushes CMD drives, but doesn't support them fully. My answer to that is that is simple: We support CMD drives in the best way possible -- by making our programs compatible, and making them run on any device number when possible. CMD designs their drives and RAM devices to be so transparent, it's difficult to think of a way to support them specifically except for partition and subdirectory surfing. I use all sorts of programs to navigate CMD devices. These programs weren't necessarily designed to be used on a CMD device.

If tbe program uses or creates files, it will either allow tbe sending of disk commands or long filenames. That's all I need for CMD support. It's one reason why I never ordered the HD version of TWS. My old version of The Write Stuff works fine on my RAMLink.

Again, LOADSTAR's main reason for pushing the FD-2000 is that one day there will *be* no 1541s in working order. That, and there will be no more new 5.25-inch disks. Can you imagine LOADSTAR falling because it's loyal readers had no way to RUN its software?

# A Commie Uprising: Internet Service Provider Revolt

*"Internet Access And You"* By: X-> MIKE <-X @ DMB The basic unit of computing is the BIT... a state determined by an ON voltage and an OFF (or null) voltage. The basic unit of networking is to transmit one bit from computer A to computer B. 8 bits make up one byte, which is equivalent to one character/letter on your screen. The 'internet' is just a collection of many networked computers. 'Internet' connectivity is highly variable, from real-time supercomputer connections to even Coca-cola machines (yep!).

The WORLD WIDE WEB is NOT the 'Internet'. It simply uses the Internet as a carrier. A good analogy is to say "my boat is the ocean". No, it is not! Your boat simply rides on the ocean.

USENET is NOT tbe 'Internet'. It also uses the Internet as a carrier.

The 'internet' is just a collection of many networked computers.

You DO NOT need 'THE WAVE' to access the 'Internet'. You MAY need 'THE WAVE' to access the GRAPHICAL portion of the WORLD WIDE WEB. You *certainly do not* need 'THE WAVE' to access the TEXTual portion of the WORLD WIDE WEB.

The WORLD WIDE WEB is just a *filesystem!* Any vt-100 capable term program should be sufficient for internet connectivity as vt-100 (and in reality, pure ascii) is the primary currency for information interchange in the known world. If your internet provider refuses to acknowledge this, find another provider.

**Q** From Caped Crusader: I don't see why we can't eventually hook up our Commodore boards with Telenet and folks could call a Commodore 8 bit board using the Internet. -----

**A:** Telnet... telEnet was (is?) an old and hopefully-now-obsolete and useless network that charged you up the wahzoo for... err, wow...slow down Mike..

I have done what you propose. I 'telnetted' a DIAL-OUT and had it call a local BBS. To be more precise, Highlander BBS in Toronto (416). Also, some guy put his BBS on the net, in a matter of speaking. He has a pc running linux acting as a go-between for the commodore. The BBS is so-so and the connection is no different than the DIAL-OUT idea (in essence you use a more pro-actively connected machine to spoon feed the commodore). What I am waiting for is an actual TCP/IP stack for the Commodore itself, which would allow it what I consider 'true' internet connectivity. It's like the difference between running a basic program and claiming you program in machine language (essentially you do, since basic is written in ML :) and actually getting down and doing ML.

PS: DO NOT ASK ME FOR A DIAL-OUT. THESE THINGS ARE SO POPULAR AND GET ABUSED THEY RARELY LAST FOR MORE THAN A FEW WEEKS.

**Q** From Realm Master: You defined the internet as bunch of "networked computers". Well, the Commodore Image network is a bunch of networked computers so would one think that it would be okay to call our CommNet an Internet? By your definition, we are an internet....one may argue that the word "internet" might mean an international network. I guess that qualifies CommNet as an Internet if for no other reason than we have some network nodes in Canada. USA-Canada..international, right? ------

**A:** God, you said that with loads o sardony... but did you realize just how much truth you were spouting out? The only differences (from a nice 'big picture' view) between our networked C=s and the "internet" are basically that links between machines are usually real-time (well...tries to be) super-fast (compared to our stuff) almost always 100% automated, and the machines tbemselves are generally serious stuff that would set you back $20 grand ez. But this is oh-so-general... Every time I connect to my IP server, in effect, my little 128 becomes part of the "internet".. please notice the caveat about "true connectivity" in the reply to CC.

XmX

# CMD Considered Including Swiftlink In Super CPU. Comments That Utility Cartridges May Work Only In 1 Mhz Mode.

**From: CMD Sales <cmd.sales@the-spa.com**
**To: H.Pieters@net.HCC.nl**
**Subject: Re: supercpu.128**

Yes, SwiftLink will be compatible with the CPU. CMD is striving to maintain compatibility with as many cartridge port peripherals as possible including RAMLink, SwiftLink, SID Symphony and the EX3/2+1 expander. As for utility cartridges such as Super Snapshot, their compatibility may be limited to 1 MHz modes because of their design and tbe fact that they take over the Kernal ROM when they are plugged in. And to your second suggestion, the answer is Yes. We did consider a built-in SwiftLink. However, it would have added too much to the units cost and would be a duplication for the many thousands of

### ADC
Add memory to Accumulator with carry
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | adc #oper | 69 | 2 | 2 |
| zero page | adc oper | 65 | 2 | 3 |
| zero page, x | adc oper,x | 75 | 2 | 4 |
| absolute | adc oper | 6D | 3 | 4 |
| absolute, x | adc oper,x | 7D | 3 | 4* |
| absolute, y | adc oper,y | 79 | 3 | 4* |
| (indirect, x) | adc (oper,x) | 61 | 2 | 6 |
| (indirect), y | adc (oper),y | 71 | 2 | 5* |

### AND
And memory to Accumulator
Flag stays affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | and #oper | 29 | 2 | 2 |
| zero page | and oper | 25 | 2 | 3 |
| zero page, x | and oper,x | 35 | 2 | 4 |
| absolute | and oper | 2D | 3 | 4 |
| absolute, x | and oper,x | 3D | 3 | 4* |
| absolute, y | and oper,y | 39 | 3 | 4* |
| (indirect, x) | and (oper,x) | 21 | 2 | 6 |
| (indirect), y | and (oper),y | 31 | 2 | 5 |

### ASL
Doubles value C ←7 6 5 4 3 2 1 0←0
Shift left one bit (memory or accumulator)
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| accumulator | asl | 0A | 1 | 2 |
| zero page | asl oper | 06 | 2 | 5 |
| zero page, x | asl oper,x | 16 | 2 | 6 |
| absolute | asl oper | 0E | 3 | 6 |
| absolute, x | asl oper,x | 1E | 3 | 7 |

### BCC
Branch to address if carry flag is clear
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bcc oper | 90 | 2 | 2* |

### BCS
Branch to address if carry flag is set
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bcs oper | B0 | 2 | 2* |

### BEQ
Branch to address if zero flag is clear
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | beq oper | F0 | 2 | 2* |

### BIT
Copy bit 7 of memory to N and bit 6 to V
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| zero page | bit oper | 24 | 2 | 3 |
| absolute | bit oper | 2C | 3 | 4 |

### BMI
Branch to address if negative flag is set
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bmi oper | 30 | 2 | 2* |

### BNE
Branch to address if zero flag not set
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bne oper | D0 | 2 | 2* |

### BPL
Branch to address if negative flag not set
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bpl oper | 10 | 2 | 2* |

### BRK
Force an interrupt (break)
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | brk | 00 | 1 | 7 |

### BVC
Branch to address if overflow flag is clear
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bvc oper | 50 | 2 | 2* |

### BVS
Branch to address if overflow flag is set
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| relative | bvs oper | 70 | 2 | 2* |

### CLC
Manually clear the carry flag
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | clc | 18 | 1 | 2 |

### CLD
Clear decimal mode
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | cld | D8 | 1 | 2 |

### CLI
Clear interrupt disable bit (enable interrupts)
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | cli | 58 | 1 | 2 |

### CLV
Clear overflow flag
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | clv | B8 | 1 | 2 |

### CMP
Compare memory with Accumulator
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | cmp #oper | C9 | 2 | 2 |
| zero page | cmp oper | C5 | 2 | 3 |
| zero page, x | cmp oper,x | D5 | 2 | 4 |
| absolute | cmp oper | CD | 3 | 4 |
| absolute, x | cmp oper,x | DD | 3 | 4* |
| absolute, y | cmp oper,y | D9 | 3 | 4* |
| (indirect, x) | cmp (oper,x) | C1 | 2 | 6 |
| (indirect), y | cmp (oper),y | D1 | 2 | 5 |

### CPX
Compare memory with X register
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | cpx #oper | E0 | 2 | 2 |
| zero page | cpx oper | E4 | 2 | 3 |
| absolute | cpx oper | EC | 3 | 4 |

### CPY
Compare memory with Y register
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | cpy #oper | C0 | 2 | 2 |
| zero page | cpy oper | C4 | 2 | 3 |
| absolute | cpy oper | CC | 3 | 4 |

### DEC
Decrement memory in address by one
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| zero page | dec oper | C6 | 2 | 5 |
| zero page, x | dec oper,x | D6 | 2 | 6 |
| absolute | dec oper | CE | 3 | 6 |
| absolute, x | dec oper,x | DE | 3 | 7 |

### DEX
Decrement X register by one
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | dex | CA | 1 | 2 |

### DEY
Decrement Y register by one
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | dey | 88 | 1 | 2 |

### EOR
Exclusive-Or W/accumulator (opposite bits on)
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | eor #oper | 49 | 2 | 2 |
| zero page | eor oper | 45 | 2 | 3 |
| zero page, x | eor oper,x | 55 | 2 | 4 |
| absolute | eor oper | 4D | 3 | 4 |
| absolute, x | eor oper,x | 5D | 3 | 4* |
| absolute, y | eor oper,y | 59 | 3 | 4* |
| (indirect, x) | eor (oper,x) | 41 | 2 | 6 |
| (indirect), y | eor (oper),y | 51 | 2 | 5* |

### INC
Increment value in address by one
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| zero page | inc oper | E6 | 2 | 5 |
| zero page, x | inc oper,x | F6 | 2 | 6 |
| absolute | inc oper | EE | 3 | 6 |
| absolute, x | inc oper,x | FE | 3 | 7 |

### INX
Increment value in X register by one
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | inx | E8 | 1 | 2 |

### INY
Increment value in Y register by one
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | iny | C8 | 1 | 2 |

### JMP
Jump to code at specified location
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| absolute | jmp oper | 4C | 3 | 3 |
| (indirect) | jmp (oper) | 6C | 3 | 5 |

### JSR
Jump to subroutine. Save return address
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| absolute | jsr oper | 20 | 3 | 6 |

*Add one cycle if page boundary is crossed. For branching, add 2. Each cycle is roughly 1 millionth of a second. In order to use these commands, you need a machine language monitor or preferably an assembler. Both of these are provided in The COMPLEAT PROGRAMMER.

# Introduction to 6510 Machine Language

by Jeffrey L. Jones. I originally wrote this article way back when I was new to assembler. Since I was closer to the moment of "eureka" then, perhaps this article is the best way to explain the mysterious commands on these two pages.

The number one reason for writing programs in machine language is speed. A program written in machine language, even if sloppy code is used, can be thousands of times faster than even a compiled BASIC program. But most people, myself included, shudder at the thought of learning how to program in machine language. After all, BASIC was tough, and now that you've mastered it, you have no desire to master another language. Well, machine language is a lot easier to learn than BASIC, especially when you already know BASIC.

Basically you're dealing with memory locations. Everything in a machine language program is concerned with "What's in this location?", "Let's move that to another location", "Let's increment", "Let's decrement..."

You have three registers, A, X, and Y. Most every operation you do will be through the ACCUMULATOR or A register. This register in your CPU is the ONLY place where mathematics can occur. It also offers the most versatile forms of ADDRESSING. More on addressing soon. Conceivably entire programs could be written without accessing the ACCUMULATOR but I think it would cause high blood pressure problems.

Let's look at a simple, unelegant machine language program that moves lines 2 and 3 to lines 0 and I on your screen. All numbers used are in decimal unless a dollar sign precedes them. Most machine language monitors would force you to use HEX, and even if they allowed you to enter decimal with a + prefix would replace the decimal with the proper hex conversion when you hit RETURN. Assemblers allow you to type in the following program as is, without the need of the line numbers. Through the magic of the fact that this ain't a real disassembly listing, you get it both ways.

```
C000:    LDY #0
C002:    LDA 1104,Y
C005:    STA 1024,Y
C008:    INY
C009:    CPY #80
C00B:    BNE $C002
C00D:    RTS
```

To write this program, you would need an assembler or a machine language monitor. You would also have to adhere to your monitor's protocol for entering the code. To RUN this program you would SYS 49152 ($C000) though it's completely relocatable since there are no JMPs or JSRs inside the code.

Here's a BASIC equivilant:

```
FORY=0TO79:A=PEEK(1104+Y):POKE1024+Y,A:NEXT
```

This program will PEEK the screen, starting at location 1104 (line 2) and then POKE it all, one by one, to line 0. The whole process takes 1.288 milliseconds to accomplish, while in BASIC it would take about two thousand times longer to execute -- *literally.*

The program starts out by LOADING the Y REGISTER with a 0 (in case it was anything else). Now we get to INDEXING: Note the command:

```
LDA 1102,Y is the same as:
A = PEEK(1102+Y)
```

This indexing is called ABSOLUTE, Y. It means you want to PEEK or POKE to Memory Location + Y

Y can be 0-255. So you can start with a base address and index to any of 256 addresses.

```
STA 1024,Y is the same as:POKE1024+Y,A
```

Again we're using ABSOLUTE,Y indexing, this time to store the data in a different location. We're MOVING 80 bytes to a location 80 bytes lower, I024 or the start of the default screen.

Next we have the commands:

```
C008:    INY
C009:    CPY #80
C00B:    BNE $C002
```

These three commands are the equivalent to *NEXT* in *BASIC.* INY increments the Y register. If Y is incremented past 255, it rolls over to zero. CPY compares Y to the number 80 or the limit-1 of our loop.

BNE $C002 is coupled with the previous command. If the comparison of Y to #80 hasn't occurred then BNE (BRANCH IF NOT EQUAL) to $C002, which is the start of the loop. The BASIC command, NEXT does the same thing: Increments a variable, checks the variable against a limit, then branches to the beginning of the loop if the limit hasn't been met.

Finally we come to RTS, which is the same as the BASIC RETURN. RTS is short for RETURN FROM SUBROUTINE. Since you will more than likely SYS to this program from the immediate mode, RTS will RETURN to the immediate mode; the same as if you GOSUB to a BASIC subroutine from the immediate mode.

Besides vastly greater speed and more compact code, machine language offers more logical operators than BASIC. With BASIC you can basically branch if:

```
equal: IF A=X THEN...                 not equal: IF A<>X THEN...
less or equal: IF A<=THEN...          less than: IF A<X THEN
greater or equal: IF A>X=THEN...      greater than: IF A>X THEN...
if not 0 (true): IF A THEN...
```

```
    But in ML you get more:
if equal.................BEQ   if zero...................BEQ
if not equal.............BNE   if not zero (true)........BNE
if less or equal.........BCC   if less than.............BCC
if greater than or equal..BCS  if greater than..........BCS
if negative..............BMI   if greater than 127.......BMI
if positive..............BPL   if less than 128..........BPL
if overflow set..........BVS   if bit 6 set (64).........BVS
if on overflow clear.....BVC   if bit 6 not set..........BVC
```

So you see that you *do* have the power to make decisions, even though you're only dealing with memory locations and numbers less than 256. Of course terms like "overflow set" are probably completely alien so I've included beneath each term what the actual logic is.

*Negative Number are all numbers greater than 127.* 0 - 127 Are Positive. 128 - 255 = Negative. So as soon as you load that accumulator (or any other register with any number, the N (negative) flag is set for you. Coupled with the N flag are two commands:

*BPL branch on plus and BMI branch on minus.* In the case of BMI $C132, the branch would occur only if the N flag was set, meaning the number in question was negative. These flags are also set by INY, DEY DEX, INX and most any command that changes a value either in memory or in a register. So when you branch based on a flag, you should be sure that the command used most recently hasn't changed any flags in ways that you don't want or expect. Usually the branch is used directly after that register is loaded or a BIT test bas been made. Check this page for details on which commands affect which flags.

Taking the previous chart into account, we can write our little line mover in a number of different ways, not to exclude counting backwards, which is my favorite way of looping.

```
C000:     LDY #79
C002:     LDA 1104,Y
C005:     STA 1024,Y
C008:     DEY
C009:     CPY #255
C00B:     BNE $C002
C00D      RTS
```

But there's something wrong with this routine. I have a totally unnecessary command embedded. I could save a minimum of 160 cycles (160 millionths of a second) and make my code shorter t'boot if I use a different form of branching. There are a few automatic boons to machine language that I haven't used yet. Again, every time you deal with a number, flags are automatically set. Let's say you LOAD the ACCUMULATOR with 35, whether you loaded it directly or PEEKed the contents of a memory location. As soon as you do it one flag is set. The ZERO flag. 35 is not equal to zero. It is also considered an unsigned "positive" number (not greater than 127). The overflow flag is set simply if the number you're dealing with has the 64 bit set. So even though 128 is greater than 64, it won't set the overflow since the 64 bit isn't set:

```
128 =  1  0  0  0  0  0  0  0
       ^  ^  ^  ^  ^  ^  ^  ^
      128 64 32 16 8  4  2  1
       N
       ^
```

FLAG TRIPPERS

As soon as you load the accumulator with 128, flags go off:

1. Not equal to zero  Z flag
2. A "negative" number:  N flag

Bearing these things in mind, let's rewrite that screen move routine:

```
C000:     LDY #79
C002:     LDA 1104,Y
C005:     STA 1024,Y
C008:     DEY;    affects N flag
C009:     BPL $C002
C00B      RTS
```

What? No comparison in the code? No need. Once Y is decremented past zero and it automatically wraps around to 255, the N flag is automatically set because Y is made greater than 127 at 255. So you KNOW that Y was just zero and you can stop.

Note that you can't use loops like this to move more than 129 bytes. This is because the loop would fall through on the first pass if Y were loaded with 129 or greater. This is because the N flag would be set after DEY since 128 is negative!



## LDA
**Load Accumulator**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | lda #oper | A9 | 2 | 2 |
| zero page | lda oper | A5 | 2 | 3 |
| zero page, x | lda oper,x | B5 | 2 | 4 |
| absolute | lda oper | AD | 3 | 4 |
| absolute, x | lda oper,x | BD | 3 | 4* |
| absolute, y | lda oper,y | B9 | 3 | 4* |
| (indirect, x) | lda (oper,x) | A1 | 2 | 6 |
| (indirect), y | lda (oper),y | B1 | 2 | 5* |

## LDX
**Load X register**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | ldx #oper | A2 | 2 | 2 |
| zero page | ldx oper | A6 | 2 | 3 |
| zero page, y | ldx oper,y | B6 | 2 | 4 |
| absolute | ldx oper | AE | 3 | 4 |
| absolute, y | ldx oper,y | BE | 3 | 4* |

## LDY
**Load Y register**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | ldy #oper | A0 | 2 | 2 |
| zero page | ldy oper | A4 | 2 | 3 |
| zero page, x | ldy oper,x | B4 | 2 | 4 |
| absolute | ldy oper | AC | 3 | 4 |
| absolute, x | ldy oper,x | BC | 3 | 4* |

## LSR
Halves value — 0 → 7 6 5 4 3 2 1 0 → C
**Shift right one bit (memory or accumulator)**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| accumulator | lsr a | 4A | 1 | 2 |
| zero page | lsr oper | 46 | 2 | 5 |
| zero page, x | lsr oper,x | 56 | 2 | 6 |
| absolute | lsr oper | 4E | 3 | 6 |
| absolute, x | lsr oper,x | 5E | 3 | 7 |

## NOP
**No Operation**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | nop | EA | 1 | 2 |

## ORA
**OR Accumulator with memory**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | ora #oper | 09 | 2 | 2 |
| zero page | ora oper | 05 | 2 | 3 |
| zero page, x | ora oper,x | 15 | 2 | 4 |
| absolute | ora oper | 0D | 3 | 4 |
| absolute, x | ora oper,x | 1D | 3 | 4* |
| absolute, y | ora oper,y | 19 | 3 | 4* |
| (indirect, x) | ora (oper,x) | 01 | 2 | 6 |
| (indirect), y | ora (oper),y | 11 | 2 | 5 |

## PHA
**Push (store) Accumulator on Stack**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | pha | 48 | 1 | 3 |

## PHP
**Push processor status on Stack**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | php | 08 | 1 | 3 |

## PLA
**Pull Accumulator From Stack**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | pla | 68 | 1 | 4 |

## PLP
**Pull processor status from stack**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | plp | 28 | 1 | 4 |

## ROL
7 6 5 4 3 2 1 0 ← C ←
**Rotate Left one bit (memory or accumulator)**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| accumulator | rol | 2A | 1 | 2 |
| zero page | rol oper | 26 | 2 | 5 |
| zero page, x | rol oper,x | 36 | 2 | 6 |
| absolute | rol oper | 2E | 3 | 6 |
| absolute, x | rol oper,x | 3E | 3 | 7 |

## ROR
→ C → 7 6 5 4 3 2 1 0
**Rotate right one bit (memory or accumulator)**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| accumulator | ror | 6A | 1 | 2 |
| zero page | ror oper | 66 | 2 | 5 |
| zero page, x | ror oper,x | 76 | 2 | 6 |
| absolute | ror oper | 6E | 3 | 6 |
| absolute, x | ror oper,x | 7E | 3 | 7 |

## RTI
**Return from interrupt**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | rti | 40 | 1 | 6 |

## RTS
**Return from subroutine**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | rts | 60 | 1 | 6 |

## SBC
**Subtract mem from accumulator with borrow**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| immediate | sbc #oper | E9 | 2 | 2 |
| zero page | sbc oper | E5 | 2 | 3 |
| zero page, x | sbc oper,x | F5 | 2 | 4 |
| absolute | sbc oper | ED | 3 | 4 |
| absolute, x | sbc oper,x | FD | 3 | 4* |
| absolute, y | sbc oper,y | F9 | 3 | 4* |
| (indirect, x) | sbc (oper,x) | E1 | 2 | 6 |
| (indirect), y | sbc (oper),y | F1 | 2 | 5* |

## SEC
**Set Carry flag**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | sec | 38 | 1 | 2 |

## SED
**Set decimal mode**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | sed | F8 | 1 | 2 |

## SEI
**Set interrupt disable status**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | sei | 78 | 1 | 2 |

## STA
**Store accumulator in memory**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| zero page | sta oper | 85 | 2 | 3 |
| zero page, x | sta oper,x | 95 | 2 | 4 |
| absolute | sta oper | 8D | 3 | 4 |
| absolute, x | sta oper,x | 9D | 3 | 5 |
| absolute, y | sta oper,y | 99 | 3 | 5 |
| (indirect, x) | sta (oper,x) | 81 | 2 | 6 |
| (indirect), y | sta (oper),y | 91 | 2 | 6 |

## STX
**Store X register in memory**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| zero page | stx oper | 86 | 2 | 3 |
| zero page, y | stx oper,y | 96 | 2 | 4 |
| absolute | stx oper | 8E | 3 | 4 |

## STY
**Store Y register in memory**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| zero page | sty oper | 84 | 2 | 3 |
| zero page, x | sty oper,x | 94 | 2 | 4 |
| absolute | sty oper | 8C | 3 | 4 |

## TAX
**Transfer Accumulator to X register**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | tax | AA | 1 | 2 |

## TAY
**Transfer Accumulator to Y register**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | tay | A8 | 1 | 2 |

## TSX
**Transfer Stack pointer to X register**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | tsx | BA | 1 | 2 |

## TXA
**Transfer X register to Accumulator**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | txa | 8A | 1 | 2 |

## TXS
**Transfer X register to Stack pointer**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | txs | 9A | 1 | 2 |

## TYA
**Transfer Y register to Accumulator**
Registers affected: N Z C I D V

| Addressing mode | Assembly form | opcode | bytes | cycles |
|---|---|---|---|---|
| implied | tya | 98 | 1 | 2 |

*Add one cycle if page boundary is crossed. For branching, add 2. Each cycle is roughly 1 millionth of a second. In order to use these commands, you need a machine language monitor or preferably an assembler. Both of these are provided in The COMPLEAT PROGRAMMER.

people who already own SwiftLink. So, we won't be including it. Thanks for your interest and I hope that we can look forward to delivering a Super CPU to you as soon as they are shipping.

CMD Sales email: cmd.sales@the-spa.com
http://www.the-spa.com/cmd/


## You, Your Commodore Computer, and the Internet, Part 5: Internet Electronic Mail

by Jim Brain. At the end of last class period, we were just delving into the simplest of all Internet services: electronic mail. Let us continue. Many of you have sent or received electronic mail, either via bulletin board systems, commercial services, or other means. Electronic mail is simply a computerized version of regular paper mail. Every mail message has the following basic parts: Sender Address, Recipient Address, Message Body.

Like regular mail, Internet electronic mail (called e-mail or email) does not contain information on how to get the message from the sender to the recipient. Also, like its more traditional counterpart, email can be addressed in a variety of ways. However, it has a number of advantages over regular mail:

There is usually no charge to send or receive email. (The exception is CompuServe & MCI, which charge per kilobyte of mail read over a certain amount. The charge is trivial, but present.) Distance between source and destination is not a factor. (Even for CIS, the charge per kilobyte is the same no matter where the message came from.) Email is almost always faster. Most email arrives in 10 minutes or less, with 1 day being the maximum delivery time for almost all messages.

It's easy to send binary files and other non textual material in email messages Sending to multiple people does not require copying of message for each individual. The paper saved is arguably more environmentally friendly.

Obviously, a few disadvantages include less personality (no special correspondence paper of fancy letterheads), computer/modem requirements, and less privacy. The last is important, as peeking at regular mail is considered a felony in many countries, but there is no such law in place for email. In fact, most email messages, if they must pass through a machine en route to their destination, are saved in areas that are easily accessible by those on that machine. Still, the sheer amount of email somewhat guarantees some element of privacy. However, if one is concerned about privacy, messages can be ciphered (just like playing detective when you were young).

Still, even with those drawbacks, Internet email is one of the most often used services of the Internet. In fact, almost all other Internet services can be accessed via email. An important advantage of email, of of the Internet in general, is its anonymity. That is, if you don't tell someone your email was sent with a Commodore computer, they'll never know. So, let's send some email.

To send a piece of email, you must know the address of the person

you are sending the message to. Just like its traditional counterpart, an email address details exactly who the intended recipient is. For example:
Traditional Internet Description
Jim BrainbrainUsername 602 N. Lemenmailmachine Fentonmsendomain MI 48430comtop-level domain.
The addresses are organized like this:
TraditionalInternet
Jim Brainbrain@mail.msen.com 602 N. Lemen Fenton, MI 48430

Just as some people have addresses that omit the street address part, some Internet email addresses have no machine part.

Because so many people ask, there is NO directory of all email addresses. There are indeed catalogs of addresses available, but they are incomplete at best. At some point, we will such an exhaustive list, but not today.

The above addressing scheme is called the "Internet" style, and is most common. However, some systems still using UUCP (Unix to Unix Copy Program) instead of TCP/IP to transfer mail might use "bang" addresses. The above address in "bang" format would be:

mail.msen.com!brain

Speaking of "bang", Internet users have a shorthand way of pronouncing some symbols. These include:

!bang *star or splat .dot @at #hash

So, the address "brain@mail.msen.com" is pronounced: brain at mail dot msen dot com. Usually, a person must give you his or her email address via some other medium. However, for large commercial services, you can infer the Internet email address if you know the username:

GEnie JIMjim@genie.com DELPHIJIMjim@delphi.com Compuserve 12345,67812345.678@compuserve.com (note ',' to '.' change) AOLjim01jim01@aol.com Prodigyjimjim@prodigy.com

Note that Internet email addresses are *not* case sensitive. jim@mail.msen.com, Jim@mail.msen.com, and JIM@MAIL.msen.com are all the same.

Now, you'll need a mail client program to send mail. For shell users, PINE is a good choice. Here is the opening screen:

```
PINE 3.91   MAIN MENU                Folder: (CLOSED)  0 Messages

    ?   HELP            - Get help using Pine

    C   COMPOSE MESSAGE - Compose and send a message

    I   FOLDER INDEX    - View messages in current folder

    L   FOLDER LIST     - Select a folder to view

    A   ADDRESS BOOK    - Update address book

    S   SETUP           - Configure or update Pine

    Q   QUIT            - Exit the Pine program

Copyright 1989-1994.  PINE is a trademark of the University of Washington.

? Help            P PrevCmd          R RelNotes
O OTHER CMDS L [ListFldrs] N NextCmd          K KBLock
```

PINE is set up as a menu interface. You can select an option by using the cursor keys, the 'P' and 'N' commands, and the letter associated with each command. For example, typing 'I' brings up the following screen:

```
PINE 3.91   FOLDER INDEX                 Folder: INBOX  Message 1 of 23 NEW

  N  1   Jan 30 Laura J. Mulcahy    (1,555) Stolen with a screwdriver
+ N  2   Jan 30 Julie Joann Brain   (1,246) Hi
  N  3   Jan 30 Eric Mercer          (728) RE: Stolen with a screwdriver
+ N  4   Jan 30 NEUS%mimi@magic.it  (4,831) Commodore Trivia Edition #25 answers
  N  5   Jan 30 Gordon Wilson       (2,422) Re: Re[2]: Is there someone???
  N  6   Jan 30 Dan Neuwirth        (1,380) Re: Stolen with a screwdriver
  N  7   Jan 30 Johnny H. Lee (EXC  (2,282) RE: Stolen with a screwdriver
  N  8   Jan 30 Kirkwood, Matt      (2,525) RE: Roof repair advice...
+ N  9   Jan 30 M.W. Cottrell       (1,451) banner for magazine
  N 10   Jan 30 TonyJess@aol.com    (1,421) Re: Roof repair advice...
  N 11   Jan 30 Laura J. Mulcahy    (1,591) Screwdriver, part II
+ N 12   Jan 30 aaron@yahoo.com     (1,322) Yahoo Change
  N 13   Jan 30 John Bolhuis        (1,923) Re: Stolen Saturns
  N 14   Jan 30 Johnny H. Lee (EXC  (2,598) RE: Roof repair advice...
  N 15   Jan 30 Machnik, Heather (  (1,844) repairing plastic panels
  N 16   Jan 30 Ray Prill           (2,938) Re: repairing plastic panels
+ N 17   Jan 30 Chris Stanford      (1,230) Comments via Mosaic
  N 18   Jan 30 Machnik, Heather (  (1,191) Screwdriver, part II
  N 19   Jan 30 Kyle Koppenhoefer   (2,045) Re: repairing plastic panels

? Help       M Main Menu  P PrevMsg     - PrevPage   D Delete     R Reply
O OTHER CMDS V View Msg   N NextMsg     Spc NextPage  U Undelete   F Forward
```

Replying to a message is as simple as selecting the message with the selection methods outlined earlier, and typing 'R'.

New users can find their way easily in PINE, but more advanced users might prefer other shell email programs. For them, elm, mh, and mail are provided.

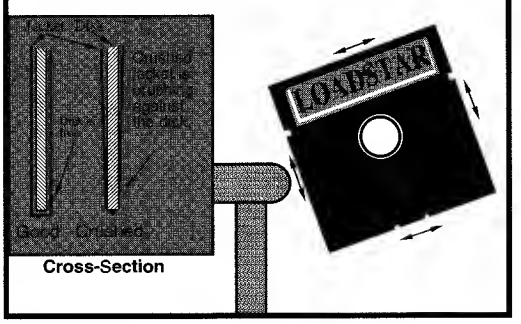
OK, we are out of time. For your homework, I would like an email message from each class member. Address the message to brain@mail.msen.com and make the Subject:

"LOADSTAR TEST MESSAGE".

Jim Brain
brain@mail.msen.com

## Disk Trouble?

Every copy of LOADSTAR leaves J & F Publishing as a verified okay disk set. 3.5-inch disks are virtually trouble-free because they are difficult to damage, even when mishandled. Unfortunately some 5.25-inch disks may become crushed in the mail and give READ ERRORS because they have trouble spinning. Fixing these crushed disks is quite easy. Just run the areas indicated along a hard edge such as a table. This should uncrimp the disk.



# CMD • HARDWARE • ORDERS: 1-800-638-3263 • SOFTWARE • CMD

## HD Series SCSI Hard Disk Drives

| | |
|---|---|
| **HD-40, 42 MB (Limited Supply)** | **$249.00** |
| **HD-500, 500+ MB (Special Edition)** | **$399.00** |
| **HD-1000, 1 GB (Special Edition)** | **$499.00** |
| **HD-2000, 2 GB (Special Edition)** | **$899.00** |

## FD Series 3.5" Floppy Disk Drives

| | |
|---|---|
| **FD-2000 (800K and 1.6 MB)** | **$179.95** |
| FD Real-Time-Clock Option/Kit | $20.00/$29.00 |
| Box of 10, High Density Disks (1.6MB) | $14.95 |
| Box of 10, Enhenced Density Disks (3.2 MB) | $29.00 |

## JiffyDOS

**(Specify computer serial number and drive model)**

| | |
|---|---|
| JiffyDOS C64/SX-64 System | $49.95 |
| JiffyDOS C-128/128-D System | $59.95 |
| Additionel Drive ROM's | $24.95 |

## Miscellaneous Hardware

| | |
|---|---|
| 80-column Monitors (Refurb) | CALL |
| Aprotek 3-Way User Port Expander | $30.00 |
| Aprotek User Port Extension Cable | $19.00 |
| C-64/64-C Computers (Refurb, w/JD) | $89.00 |
| C-64, C-64C Power Supply (Repairable) | $39.00 |
| C-128 Power Supply (Repairable) | CALL |
| Cennon BJC-4100 Color Bubble Jet Printer | $399.00 |
| Commodore 1541 Disk Drives (Refurb, w/JD) | $75.00 |
| Commodore 1541-II Disk Drive (Refurb, w/JD) | $89.00 |
| Commodore 1571 Disk Drive (Refurb, w/ JD) | $99.00 |
| Commodore 1802 40-column Monitor (Refurb) | $149.00 |
| CMD EX2+1 3-Port Cartridge Port Expander | $34.95 |
| CMD EX3 3-Port Cartridge Port Expander | $29.95 |
| CMD Gamepad/Joystick Controller | $24.95 |
| CMD GeoCable II Print Cable 6ft./15ft. | $29.00/$34.00 |
| CMD SmartMouse (1351 Compatible Mouse) | $49.95 |
| CMD SmartTreck (1351 Competible Trackball) | $69.95 |
| Inkwell Light Pen Model 170C | $75.00 |
| Monitor Cables | CALL |
| Mouse Pad | $2.95 |
| MW-350 Printer Interface (0K/8K Buffer) | $49.00/$60.00 |
| Panasonic KXP-1150 9-pin Printer | $169.00 |
| Panasonic KXP-2023 24-pin Printer | $229.00 |
| Panasonic KXP-2135 24-pin Color Printer | $319.00 |

## Utilities

| | |
|---|---|
| Big Blue Reader V4.10 (SOGWAP) | $39.00 |
| CMD Utilities | $24.95 |
| JiffyMON-64 (ML Monitor) | $19.95 |
| The Compression Kit '94 (Mad Men) | $39.00 |

## GEOS

| | |
|---|---|
| Collette Utilities (Handy Geos Utilities) | $19.95 |
| Desk Peck Plus | $29.00 |
| Oweezils Greetest Hits (NewTools2,Stamp,Label128) | $30.00 |
| FONTPACK Plus | $25.00 |
| geteWey 64 or 128 (Specify Version) | $29.95 |
| geo8ASIC | $20.00 |
| geoCelc 64/128 | $40.00/$45.00 |
| geoChert | $29.00 |
| geoFAX | $39.95 |
| geoFile 64/128 | $40.00/$45.00 |
| geoMekeBoot (Mekes Booteble copies) | $12.95 |
| geoProgrammer | $45.00 |
| geoPublish | $40.00 |
| GEOS 64 v2.0 | $44.00 |
| GEOS 128 v2.0 | $49.00 |
| geoSHELL V2.2 (CLI for GEOS) | $24.95 |
| Internetionel FONTPACK | $25.00 |
| Perfect Print LQ for GEOS (Laser-like output) | $49.95 |
| RUN GEOS Companion | $20.00 |
| RUN GEOS Power Pak I or II (Specify) | $20.00 |

## Books

| | |
|---|---|
| Anatomy of the 1541 | $12.00 |
| C-64 Science & Engineering | $12.00 |
| C128 Computer Aided Design | $12.00 |
| C128 BASIC Training Guide | $12.00 |
| Commodore 64 Tricks and Tips | $12.00 |
| GEOS Programmers Reference Guide | $35.00 |
| Graphics Book for the C-64 | $12.00 |
| Hitchhikers Guide to GEOS | $35.00 |
| Ideas for Use on Your C-64 | $12.00 |
| Printer Book for the C64 | $12.00 |
| Mapping the C64 | $12.00 |
| Simple Internet | $16.95 |
| Superbese - The Book | $15.00 |

## Scanning & Video

| | |
|---|---|
| Hendyscanner | $249.00 |
| Pagefox | $139.00 |
| **Video Digitizer** | **$189.00** |
| **Vidaofox** | **$89.00** |

## RAMLink Power-Backed RAM Disk

| | |
|---|---|
| RAMLink Bese Model (ØMB, No RAMCard) | $149.00 |
| **RAMLink w/1 MB RAMCard (Special!)** | **$199.00** |
| **RAMLink w/4 MB RAMCard (Special!)** | **$299.00** |
| RAMCerd RTC Option/Kit | $10.00/$19.00 |
| RAMLink Bettery Back-up (Optional) | $24.95 |
| Perallel Cable (RAMLink to HO) | $14.95 |

## MIDI & Sound

| | |
|---|---|
| Oigimaster | $34.95 |
| SID Symphony Stereo Cartridge | $44.95 |
| Sonus 64 Sequencer, MIOI Interfece, Cebles | $99.00 |
| Sonus 128 Sequencer, MIDI Interfece, Cables | $99.00 |

## Languages & Compilers

| | |
|---|---|
| BASIC 64 Compiler (Abacus) | $17.00 |
| BASIC 128 Compiler (Abecus) | $25.00 |
| Blitzl 64 Compiler (Skyles) | $30.00 |
| Blitzl 128 Compiler (Skyles) | $30.00 |
| Buddy 64/128 Assembler | $39.00 |
| Cobol 64 (Abacus) | $17.00 |
| Fortran 64 (Abacus) | $17.00 |
| Pascal 64 (Abacus) | $17.00 |
| Power C 64 (Spinnaker) | $14.00 |

## Productivity

| | |
|---|---|
| Cedpek 64 (Abacus) | $22.00 |
| Cedpak 128 (Abacus) | $25.00 |
| Chartpak 64 (Abacus) | $17.00 |
| Chartpak 128 (Abacus) | $25.00 |
| **Data Manager 64 (Timeworks)** | **$16.00** |
| I Peint v1.5 (128, 80-col., 64K VDC) (Living Proof) | $39.00 |
| I Port v1.54 (128, 80-col., 64K VDC) (Living Proof) | $29.00 |
| Personal Portfolio Manager (Abacus) | $16.00 |
| Pocket Writer 1 (64) Digital Sol.) | $35.00 |
| Pocket Writer 2 (64/128) (Oigital Sol.) | $65.00 |
| Pocket Writer 3 (64 or 128) (Digital Sol.) | $70.00 |
| Pocket Planner 2 or Pocket Filer 2 (Oigitel Sol.) | $35.00 |
| RUN Productivity Pak I, II, or III (Specify) | $15.00 |
| RUN Super Sterter Pek 1541 or 1581 | $20.00 |
| RUN Works | $20.00 |
| SEC Check Register 128 | $29.00 |
| SuperScript 64 (Precision) | $15.00 |
| SuperScript 128 (Precision) | $20.00 |
| Suberbese 64 Version 3.01 (Precision) | $35.00 |
| Superbase 128 Version 3.01 (Precision) | $35.00 |
| SwiftCelc 64 (Timeworks) | $16.00 |
| Perfect Tax 64 (Free '95 Upgrade) | $69.00 |
| Perfect Tax 128 (Free '95 Upgrade) | $79.00 |
| TWS 64 w/Speller (Busy Bee) | $29.00 |
| TWS 128 w/Speller (Busy Bee) | $39.00 |
| TWS Modules (HD/RL/Illustrator) | each $5.00 |

## Telecommunications

| | |
|---|---|
| Aprotek MiniModem C-24 (C= ready, 2400 baud) | $69.00 |
| Aprotek MiniModem C (C= ready, 1200 baud) | $50.00 |
| Aprotek Modem adapter (C= to Ext. PC Modem) | $20.00 |
| BOCA 2400 Baud Modem | $69.00 |
| BOCA 2400 w/SwiftLink and Cable | $99.00 |
| BOCA 14.4K bps FaxModem | $129.00 |
| BOCA 14.4K w/SwiftLink & Cable | $159.00 |
| BOCA V.34 28.8K bps FaxModem | $199.00 |
| BOCA V.34 w/SwiftLink & Cable | $229.00 |
| Oialogue 128 | $29.00 |
| SwiftLink RS-232 Cartridge (Up to 38.4K baud) | $39.95 |
| SwiftLink Modem Cable (OB9-DB25) | $9.95 |

## Games

| | |
|---|---|
| Atomino | $17.00 |
| Ballistix | $13.00 |
| Blood Money | $15.00 |
| Chompl | $10.00 |
| Oay in the Life of Prehistoric Man | $19.00 |
| Escape Route | $19.00 |
| Heevenbound | $19.95 |
| Island of the Oregon | $19.00 |
| Laser Squed | $13.00 |
| **Lazer Duel** | **$14.95** |
| Lions of the Universe | $19.95 |
| Mainframe | $13.00 |
| Menace | $15.00 |
| Navy Seal | $10.00 |
| Rings of Medusa | $16.00 |
| RUN C128 Funpak | $10.00 |
| RUN C64 Gemepek | $10.00 |
| Skete or Die | $13.00 |
| **Sletarman** | **$14.95** |
| The Amazing Spider-Men | $15.00 |
| The President Is Missing! | $10.00 |
| The Three Stooges | $10.00 |
| Tie Breek Tennis | $16.00 |
| Total Eclipse | $10.00 |
| Ultima V | $17.00 |
| **Waikarz** | **$19.00** |
| Wings of Circe | $19.00 |
| Wizerdry 5: Heart of the Meelstrom | $24.00 |

## JiffyDOS

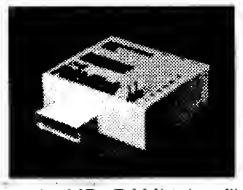
**Increase Speed Up to 1500% while retaining 100% compatibility**

- Speeds up Loading, Saving, Verifying, Formatting and Reading/ Writing of Program, Sequential, User and Relative files unlike cartridges which only speed up Loading and Saving of PRG files
- Built-in DOS Wedge plus 17 additional features including file copier, text dump, printer toggle, and redefinable function keys make using your computer easier and more convenient
- ROM upgrade installs easily into most computers and disk drives. Supports C-64, 64C, SX-64, C-128, 128-D, 1541, 1541C, 1541-II,1571, 1581 and more.

## FD Series

**1.6 MB 3.5" Floppy Drives**

The FD-2000 disk drives utilize today's latest 3.5 inch disk drive technology. FD-2000's support 800K (1581 style) and 1.6 MB (High Density) formats. Fast and reliable, they support 1541, 1571 and 1581 style partitions, Native Mode partitioning and can actually read and write 1581 disks. FD's feature built-in JiffyDOS, SWAP button and optional Real Time Clock. High capacity, speed and compatibility make the FD right for every application, including GEOS.



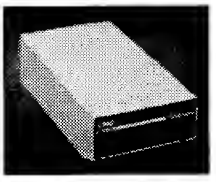
## RAMLink

**Power-Backed Expandable REU**

The fastest possible form of storage, RAMLink provides instant access to files and retains data while your computer is turned off. Easy to use and expandable up to 16 MB, RAMLink utilizes the same powerful operating system found in CMD Hard Drives. Unlike Commodore REU's which are compatible with less than 1% of commercial software, RAMLink supports more than 80% of the commercial titles. RAMLink also offers built-in JiffyDOS Kernal, SWAP feature, reset button, enable/ disable switch, pass-thru port and RAM port for REU/GEORAM use. RAMLink offers maximum speed, expandability and compatibility with all types of software and hardware including GEOS.

## HD Series

**SCSI Hard Drive for the C-64/128**



HD Series Hard Drives are available in capacities up to 1 GB, are fully partitionable, and can emulate 1541, 1571, & 1581 disks while Native partitions utilize MS-DOS-style subdirectories. HD's connect easily to the serial bus or parallel via RAMLink. Includes built-in JiffyDOS, SWAP feature and RTC. HD's offer superior compatibility with most commercial software including BBS, Productivity and GEOS. And with new pricing, HD Series drives offer the lowest cost per megabyte of any C64/128 storage device.

## Shipping and Handling Charges

Use the chart below to match your order subtotal with your shipping zone and method

| | Continental United States | | | | |
|---|---|---|---|---|---|
| **Order Subtotal** | **UPS Ground** | **2nd Day** | **Next Day** | **AK HI, PR** | **Canada** |
| $0.01 - $19.99 | $3.00 | $8.00 | $17.00 | $12.00 | $5.00 |
| $20.00-$29.99 | $5.00 | $9.00 | $18.00 | $14.00 | $7.00 |
| $30.00-$59.99 | $6.00 | $10.00 | $20.00 | $15.00 | $9.00 |
| $60.00-$149.99 | $6.00 | $12.00 | $23.00 | $19.00 | $12.00 |
| $150.00-$299.99 | $10.00 | $14.00 | $27.00 | $21.00 | $20.00 |
| $300.00-$799.99 | $15.00 | $20.00 | $34.00 | $27.00 | $25.00 |
| $800.00 + | $20.00 | $25.00 | $40.00 | $32.00 | $35.00 |

UPS C.O.D. add $5.00 (U.S./PR only)

**Policies:** CMO accepts: Major Credit Cerds, Money Orders, COO end Personal Checks. Personel Checks are held 3 weeks. Customer is responsible for shipping cherges on refusals. Most items ere stock, contact CMO for firm delivery. Returns for merchandise credit only within 30 days with prior authorizetion. Credits ere less shipping, handling, texes, customs, duties, end a 10% restocking cherge. No refunds or credits on opened softwere. All prices end specificetions ere subject to chenge without notice.

MasterCard VISA **WE ACCEPT ALL MAJOR CREDIT CARDS** DISCOVER NOVUS


**Creative Micro Designs, Inc.**
P.O. Box 646
East Longmeadow, MA 01028
**Info:** (413) 525-0023
**Fax:** (413) 525-0147